

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings of claims in the application:

Listing of Claims:

1 1. (Currently Amended) A method of optimizing a query in a multi-
2 tenant database, said multi-tenant database having one or more data tables, each table
3 having one or more logical columns defining data categories and one or more logical
4 rows associated with one or more tenants, wherein a plurality of tenants have data stored
5 in the data tables, the method comprising:

6 generating tenant-level statistics for ~~each of said~~ one or more of said
7 plurality of tenants for ~~each of~~ one or more of the data tables;

8 receiving a SQL query; and

9 optimizing the SQL query based on the tenant-level statistics.

1 2. (Currently Amended) The method of claim 1, wherein each tenant
2 includes one or more associated users, the method further including:

3 generating user-level statistics for ~~each user~~ one or more of the users of
4 ~~each tenant~~ one or more of the tenants for ~~each~~ one or more of the data tables; and

5 optimizing the SQL query based on the user-level statistics.

1 3. (Original) The method of claim 2, wherein the user-level statistics are
2 stored to a user metadata table.

1 4. (Original) The method of claim 2, wherein generating user-level
2 statistics includes determining a total number of distinct rows for each of said plurality
3 of users.

1 5. (Original) The method of claim 4, wherein the total number is an
2 approximate number based on one or more of a) a number of rows viewable by the user

3 and users below the user in a role hierarchy, b) a number of rows that are shared by a
4 group to which the user belongs and c) a number of rows that are manually shared to the
5 user by another user or group of users.

1 6. (Original) The method of claim 2, wherein generating user-level
2 statistics for a user is performed according to one of a) on a scheduled basis, b) after a
3 predetermined number of queries by the user, and c) each time an unconstrained query is
4 run by the user.

1 7. (Original) The method of claim 1, wherein generating tenant-level
2 statistics is performed on a periodic basis.

1 8. (Original) The method of claim 1, wherein generating includes
2 determining a total number of distinct rows accessible for each of said plurality of
3 tenants.

1 9. (Original) The method of claim 8, wherein the tenant-level statistics
2 are stored to a tenant metadata table.

1 10. (Original) The method of claim 1, wherein at least one column of one
2 of said tables includes data associated with two or more tenants.

1 11. (Currently Amended) A multi-tenant database system, comprising:
2 a database having one or more data tables, each table having one or more
3 columns defining data categories and one or more rows associated with one or more
4 tenants, wherein a plurality of tenants have data stored in the data tables;
5 a statistics generating module configured to generate tenant-level statistics
6 for ~~each tenant~~ one or more tenants for ~~each~~ one or more of the data tables; and
7 a query optimization module, configured to optimize a database query
8 based on the tenant-level statistics.

1 12. (Original) The multi-tenant database system of claim 11, wherein

2 each tenant includes one or more associated users, wherein the statistics generating
3 module is further configured to generate user-level statistics for each user, and
4 wherein the query optimization module is further configured to optimize the
5 database query based on the user-level statistics.

1 13. (Original) The system of claim 12, further including a memory
2 module, wherein the statistics generating module stores the user-level statistics to a
3 metadata table in the memory module.

1 14. (Original) The system of claim 12, wherein the statistics generating module
2 generates user-level statistics by determining a total number of distinct rows for each of said
3 plurality of users.

1 15. (Original) The system of claim 14, wherein the total number is an
2 approximate number based on one or more of a) a number of rows viewable by the
3 user and users below the user in a role hierarchy, b) a number of rows that are shared
4 by a group to which the user belongs and c) a number of rows that are manually
5 shared to the user by another user or group of users.

1 16. (Original) The system of claim 12, wherein the statistics generating
2 module generates user-level statistics for a user according to one of a) on a scheduled
3 basis, b) after a predetermined number of queries by the user, and c) each time an
4 unconstrained query is run by the user.

1 17. (Original) The system of claim 11, wherein the statistics generating
2 module generates tenant-level statistics on a periodic basis.

1 18. (Original) The system of claim 11, wherein the statistics generating module
2 generates tenant-level statistics by determining a total number of distinct rows viewable for
3 each of said plurality of tenants.

1 19. (Original) The system of claim 18, further including a memory module,

2 wherein the statistics generating module stores the tenant-level statistics to a tenant
3 metadata table in the memory module.

1 20. (Original) The system of claim 11, wherein at least one column of one
2 of said tables includes data associated with two or more tenants.

1 21. (Currently Amended) A method of optimizing a query in a multi-
2 tenant database, said database having one or more data tables, each table having one
3 or more logical columns defining data categories and one or more logical rows
4 associated with one or more tenants, wherein a plurality of tenants have data stored in
5 the data tables, and wherein each tenant includes one or more users, the method
6 comprising:

7 processing the data tables so as to determine tenant-level statistics
8 for each of said plurality of tenants;

9 processing the data tables so as to determine user-level statistics for
10 each of said plurality of ~~user~~ users;

11 receiving a SQL query; and

12 optimizing the SQL query based on one or both of the tenant-level
13 statistics and the user-level statistics.

1 22. (Original) The method of claim 21, further including:

2 storing the user-level statistics to a user-level metadata table in a
3 memory module; and

4 storing the tenant-level statistics to a tenant-level metadata table in
5 the memory module.

1 23. (Original) The method of claim 21, wherein determining user-level
2 statistics includes determining a total number of distinct rows for each of said
3 plurality of users, and wherein determining tenant-level statistics includes
4 determining a total number of distinct rows for each of said plurality of tenants.

1 24. (Original) The method of claim 21, wherein processing the data tables
2 to determine tenant-level statistics is performed on a periodic basis.

1 25. (Original) The method of claim 21, wherein processing the data tables
2 to determine user-level statistics for a user is performed according to one of a) on a
3 scheduled basis, b) after a predetermined number of queries by the user, and c) each
4 time an unconstrained query is run by the user.

1 26. (New) The method of claim 1, wherein the generating tenant-level
2 statistics comprises generating tenant-level statistics for each of said plurality of
3 tenants for each of the data tables.

1 27. (New) The method of claim 2, wherein the generating user-level
2 statistics comprises generating user-level statistics for each of the users of each of the
3 tenants for each of the data tables.

1 28. (New) The multi-tenant database system of claim 11, wherein the statistics
2 generating module is configured to generate tenant-level statistics for each tenant for each of the
3 data tables.